“It’s Not Exactly Meant to Be Realistic”: Student Perspectives on the Role of Ethics In Computing Group Projects

Michelle Tran
michelle.h.tran@colorado.edu
University of Colorado Boulder
Boulder, CO, USA

Casey Fiesler
casey.fiesler@colorado.edu
University of Colorado Boulder
Boulder, CO, USA

ABSTRACT

In computing education, group projects are often seen as an opportunity for students to gain experiences similar to what they will face in the workplace. As such, it is a pressing area for ethics education: practice incorporating ethics into project-based courses might provide encouragement and scaffolding for students to consider ethics when working on “real” technologies. In this work, we provide a preliminary look at the state of ethics education in project-based computing courses via semi-structured focus groups. These focus groups revealed that while not completely ignored, ethics had a very small role in our participants’ group projects, especially in software engineering courses. Furthermore, participants generally agreed that while group projects can be useful learning opportunities, they are not realistic, and this impacts their willingness to consider ethical implications while working on projects. We compiled participant feedback and ideas on how to improve ethics education in computing group projects, and present a synthesis on how this feedback can be implemented in the classroom. We also emphasize that part of simulating "real" tech work should include acknowledgment of larger structural problems and ideally training and practice in communication and argumentation.

CCS CONCEPTS

• **Social and professional topics**;

KEYWORDS

ethics, computing education, group projects, university

ACM Reference Format:
Michelle Tran and Casey Fiesler. 2024. “It’s Not Exactly Meant to Be Realistic”: Student Perspectives on the Role of Ethics In Computing Group Projects. In *ACM Conference on International Computing Education Research V.1 (ICER ’24 Vol. 1), August 13–15, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3632620.3671109

1 INTRODUCTION

A growing “techlash" fueled by ethical controversies and evidence of societal harms has led to demand for increased ethical consideration during the technology development process [28, 46]. However, at the same time, many people working in the tech industry report challenges in navigating ethical issues that arise [19, 48]. In response, higher education has become more focused on making ethics and related concepts (such as responsibility and justice) a more integral part of computing education [13]. However, there are many opinions on how to best approach ethics education, and the computing community is still settling on appropriate pedagogy and strategies [45]. As a result, attention to ethics education has increased dramatically in recent years, with a variety of interventions being trialed and evaluated [8, 38].

Presumably, this growing body of work on ethics education is aimed at producing graduates who will be able to take these skills and knowledge into the real world. However, many of these interventions focus on the pedagogical goals of issue spotting and general ethical awareness [13, 38], which, though important first steps, are unlikely to adequately prepare students for the complex realities of technology development, such as organizational incentives and issues of power [48]. Therefore, in this project, we take a step toward what ethics education targeted explicitly on *practice* might look like, by focusing on the component of computing education that is often designed to mimic real world settings: group projects.

Project-based courses are the current standard for simulating “real world” tech industry experiences. These courses are often designed to simulate the real software development process [1, 2, 29], and the typical goal of capstone courses specifically is to prepare students to succeed in the "real world" [35]. The absence of ethical considerations in project-based courses therefore risks implying to students that they can also work on “real” software without these considerations – so this is a particularly pressing area to consider the potential for ethics pedagogy.

In order to explore the challenges, opportunities, and potential strategies for this type of curricular integration, this preliminary investigation on the role of ethics in computing group projects focuses on the following exploratory research questions, based on students at a single university:

(1) Do, and in what ways do, students working on technical group projects consider the ethical ramifications of their work?
(2) How have students working on group projects been encouraged and/or supported by their course to consider ethics in their projects, and to what extent have those methods been effective?
(3) What kind of support or scaffolding would students find useful for incorporating ethics into their design process during group projects?

ICER ’24 Vol. 1, August 13–15, 2024, Melbourne, VIC, Australia
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0475-8/24/08
https://doi.org/10.1145/3632620.3671109

We conducted focus groups with computing students currently working on group projects about whether they consider ethics in their work, what ethical support they currently receive in their courses, and what kinds of support they would find helpful during their projects. These focus groups revealed that while not completely ignored, ethics had a very small role in our participants' group projects, especially in software engineering courses. Furthermore, participants generally agreed that while group projects can be useful learning opportunities, they did not see them as realistic, which impacted their willingness to consider ethics while working on projects. We compiled participant feedback and ideas on the role of ethics in computing group projects, and present suggestions for classroom implementations based on their feedback and prior work on technology ethics education and practice.

2 BACKGROUND AND RELATED WORK

When referencing the role of "ethics" in computing education, it is important to note that this term often does not have a clear definition even in the literature [8], and may apply to a collection of related concepts such as socially responsible computing [10], critical computing [27], or justice-centered approaches [30]. As noted in a recent analysis of educator attitudes toward ethics education, people's definition of "ethics" can range from "doing the right thing" to explicitly making injustices visible [45]. Similar to that work, in this project we left the term intentionally open-ended, and used related terms such as "impact" when questioning students. However, in the computing research community, "ethics" is a common term for this area of work.

A growing body of literature within computer science education reports on strategies for ethics education in computing [8]. For example, research around integrating applied ethics into computing classes has shown promise for traditional, problem-set focused classes, demonstrating the viability of techniques such as repurposing existing assignments to discuss technical topics in the context of an ethical problem, or adding ethics-focused discussions in order to increase student engagement and critical thinking [7, 12, 20, 43]. Evaluative research on ethics education interventions has also shown that embedding ethical discussion into technical computing classes can increase student interest in tech ethics, as well as their confidence in identifying and discussing ethical issues [23]. Additionally, a long-term evaluation of embedded ethics education suggested that once students were in the workforce, those who had had ethics modules as part of their training had increased ethical sensitivity and were more likely to report having encountered ethical issues in their jobs – however, less than half of those same former students reported that their training had been explicitly helpful when they encountered those issues [22].

It is very possible that common learning objectives for ethics education such as issue spotting and general awareness may not adequately prepare students for the realities of encountering ethical issues in industry settings [19, 48]. They also tend to focus on students' roles and decisions as individuals, rather than encouraging a critical focus on structural issues [25].

One pilot program designed towards introducing computing students to realistic work scenarios is the Platform for Ethics and Responsible Computing Education (PEaRCE), an online learning environment where students take on the role of employees within simulated projects and have the opportunity to speak to simulated stakeholders about the potential impacts of those projects [9]. One component of this simulation is to introduce realistic constraints such as time sensitive decision-making in a real workplace, and the pilot suggested that this system is a promising direction for helping to prepare students for real-world ethical scenarios in industry roles.

Indeed, a survey of faculty regarding their views of undergraduate computer science education revealed that a majority believed that the primary goal of undergraduate education was to prepare students for industry roles, and that their university would benefit from investigating how to better prepare students for those industry careers [47]. When it comes to preparing for careers after a computing degree, group projects are commonly considered to be the closest opportunity students have to understanding how the software development process works [1, 2, 29]. For capstone courses specifically, a common learning objective is for students to "demonstrate the ability to succeed in a software design experience similar to the real world" [35]. In some fields, there has been explicit discussion of the role of ethics in such courses–for example, the acknowledgment that topics such as social justice, bias, and inclusion in data and design are critical for bioengineering students [18]. Such topics are also common in standalone computing ethics courses, which frequently include discussions of topics like surveillance, inequality, bias, and civic responsibility [13]. However, as observed by Bielefeldt et al in 2017, computing capstone projects incorporate ethics at a far lower rate than several other engineering disciplines: only 10% of computer science respondents stated that their capstone projects incorporated ethics, though it is possible that figure has changed given that particular survey dataset was collected in 2005 [3].

The same survey about faculty views of computer science undergraduate education also revealed that teaching communication skills and teaching about social context were both very commonly cited as ways to improve student preparation for the workforce, ahead of even teaching more engineering skills [47]. Additionally, prior work has suggested that ethical components to computing assignments can actually make them feel more applicable to the "real world" for students [24, 33]. We therefore began this investigation to explore whether integrating ethics and related concepts as explicit components of group projects might be a path toward better preparing students for real world ethical scenarios that might arise once they are in the workforce.

3 METHODS

We conducted 11 hour-long semi-structured focus groups with teams of students already working on computing projects together. Focus group participants were recruited through physical posters hung up in areas where computing students would typically study, as well as digital posters distributed through professors via lecture slides, university listservs, and private university forums for computing students. Participants were eligible as long as they were at least 18 years old and currently working on a team project in a computing-related course at our university, where at least one other person on the team was willing to attend. Participants were compensated with a $30 gift card.

The first author conducted these focus groups, where the initial questions were derived from our research questions, and focused on understanding what participants were working on and what kind of support they received in thinking about the ethical implications of their projects, as well as gauging how much they had thought about these topics on their own, and what additional support they could use. Examples of these initial questions include: "Throughout your team’s work on the project, from deciding on project ideas all the way to implementing it, have you thought about the positive and negative impacts of this work?" and "How does your project course support you in thinking about, preventing, and/or mitigating potential harms your software might cause?"

For each question, participants silently contributed individual ideas on a Google Jam Board. Then, the team would discuss their responses to the question. Similar to semi-structured interviews that allow for the flexibility to adapt to participant responses [41], the researcher also asked follow-up questions. As noted previously with respect to multiple interpretations of the term "ethics," we first asked more general questions around their projects and questions of considering "impact," prior to using the more loaded term "ethics."

This study was approved by our university’s institutional review board, and focus groups were conducted during the latter half of the Spring 2023 semester. The 11 focus groups consisted of teams of 2-6, with 29 participants total. Ages ranged from 19-39, with over 95% of participants falling between 19 and 27. Roughly two thirds were graduate students, and the other third were undergraduates. The vast majority were computer science majors; the 3 students who were not pursing computer science were studying other types of engineering, such as electrical and mechanical. Approximately 40% were women, and 60% were men. Over 75% of participants identified as Asian, with about 10% white, and the rest identifying in another way.

At the large state research university where this research was conducted, the population is predominantly white, and there are both undergraduate and graduate students. Ethics curricula and requirements vary across relevant majors at the university, but because most participants in our study were computer science majors, it is worth noting that undergraduates in that major are required to take an ethics course (with the option to take a technology-focused course or a general ethics course in the philosophy department). The computer science program is also ABET accredited, which requires incorporating computing ethics topics in both introductory and capstone courses. However, many of our participants were graduate students, which means that they may have received their undergraduate training elsewhere. For their graduate studies, computing MS and PhD students have technology ethics courses available as electives, but they are not requirements.

Following data collection, we conducted thematic analysis [4] on the transcripts. Both authors conducted open coding by highlighting noteworthy sections and categorizing ideas based on the focus group question they were responding to (and consequently the research question the sentiment addressed), broad topics that the participant touched on, and then subtopics within those broad topics that indicated student sentiments (i.e. if the broad topic was about how prepared students felt to consider ethics in computing projects, the subtopics could include remarks stating that they felt prepared, or statements where students noted they did not feel prepared). We then collectively reviewed common sentiments to draw fewer than 15 themes from the coding, which were further discussed, consolidated, and iterated upon between both authors. The first author then drafted theme memos that were further collectively iterated on. We discuss these themes below in our findings, using participant quotes that represent broader themes. Participants are labeled as PX-Y, where X is the group they belonged to, and Y is the particular participant within the group. Quotes have only been edited for clarity.

3.1 Limitations

This study was conducted on a narrow sample, and given that the 29 participants were invited as 11 established groups, there are a few things to keep in mind while interpreting this work’s findings. First, while the inclusion of graduate and undergraduate student groups allows us to derive a greater breadth of insight into advanced computing coursework, it also adds more volatility to the sample, which could create additional noise. This also means that the students have not had uniform experiences with ethics education (since most of the graduate students studied computing fields for their undergraduate education at other universities). Groups also came from classes where their approach to the group project varied significantly – for example, some classes randomly assigned project partners while others allowed students to choose their groups, which may have resulted in very different social dynamics between groups. There was also a significant difference between the sample’s demographics vs. the overall institution’s demographics, which could affect how representative it is of even this single institution’s student body. However, our goal with this study was not for the results to be generalizable, but to provide a starting place for exploring challenges and opportunities in this space.

4 FINDINGS

Participants shared their experiences around several key topics, including perceived project realism, the ethics education they had received so far (if any), and what they might find helpful for learning how to consider ethics for their projects.

4.1 Group Projects vs the Real World

A common sentiment among participants was that group projects did not seem “real.” As P1-2 expressed: group projects are *“not exactly meant to be realistic.”* This lack of realism impacted how they perceived the importance of ethical considerations in these projects.

Participants tended to see their group projects as a sandbox where they would not interact with real users, thus there was no risk of impacting people and society. Without the risk of real impact, they did not think it would be useful to stop and think about possible ethical ramifications. P3-2 used the number of real users as a metric for whether thinking about ethics would be important, stating:

> *“The main thing is, I feel like the user base won’t be that big. Because it’s a class project, and we’re not really planning on putting it on the internet or anything.*

> *So, the people that'd be seeing it would be us and our professor, and maybe one or two of our classmates. [...] It doesn't feel like it has that big of an impact."*

In addition to having very few users, participants noted the lack of external stakeholders also reduced impact. When reflecting on the effect their project could have on people, P7-1 noted that *"We don't really have any user requirements... we haven't interviewed any users."* Not having "real" user requirements or stakeholders divorced the project from real world considerations.

Other participants noted that the lack of alignment with their perceptions of the tech industry and its practices contributed to whether they thought about ethics in their projects. Many participants expressed that the purpose of pursuing their degree was so they could get a job, and their outlook on industry impacted what they thought they should focus on and learn. Corroborating a finding from prior work about ethics and job hunting [40], cynicism about tech companies was a common reason for why participants did not think tech companies would value the ability to think about ethics as a skill. For example, P10-3 stated their opinion that tech companies were actively trying to have *"negative impact,"* giving the example of pushing for *"highly addictive behaviors"* among users. Participants also did not think software engineers would have the power to prevent unethical software from being produced by the company, which tracks to research findings about the attitudes of working tech professionals [48]. P11-1 stated:

> *"If it's only the software engineer who's working on the project, then [they] have the control to [think about] negative impact. But in reality, there are also project managers [...] And other engineers, or SWE managers, and also the top level of the company [who] will engage to design products [...] it's still possible to raise up the question about the impact of the project, but it might be hard to [get] consent, because in such a large company, you might not have [the] right to change the product."*

This idea that projects do not reflect reality caused participants to skip the step of ethical assessment. Many stated they had not thought about ethics in this context before participating in our study. As P10-5 stated: *"I'd never thought about it because mainly I thought [...] it's a school project and no offense, I don't think it's gonna be online."* Those who did think about impact often thought of just the positive side. P8-1 reflected on their positive-lean during development, as well as their perceptions of industry: *"We had positive impacts in mind, we didn't think about [the] negative. I think that is generally [the case] in any industry or any company."* Several participants expressed that they appreciated the chance to think about the ethical ramifications of their work during our study; P6-1 thanked the researchers for bringing up the topic of ethics in tech, stating: *"You are helping us. The one good point is we hadn't thought about it that much, but you're making us question all this stuff, [...] which will definitely help us."*

While group projects were not perceived as "realistic," participants still noted that they did prepare them for the technical and social aspects of industry to some degree – but not for creating ethical tech. When asked what group projects were useful for, P9-2 noted the technical skill benefits:

> *"This project that we are doing [...] is just for our quest to learn some new technologies and new tools that are available in the market to expose ourselves to a real world tech stack that [is] available."*

P9-1 noted the behavioral teachings of computing group processes and working on a team, saying

> *"In collaboration among different team members, how do you collaborate? So for example, if I'm working on something and [P9-2] is dependent on that work, I need to prioritize that first so that [they're] unblocked and [they] can continue [their] work further."*

Yet, when asked whether they would feel prepared to sit down and think about the possible ethical ramifications of any software they produce, participants felt much less certain. P6-1 and P6-2 replied *"No"* and *"I guess not"* almost immediately, while P11-2 suggested that they would have to do some work to feel prepared: *"I think I would've done some research."*

4.2 Perceptions of Ethics Education

Because many of our participants were graduate students who had completed their undergraduate coursework at other universities, their own experiences with computing ethics education varied widely, and so did their attitudes toward it. Only about a third of the participants had taken a dedicated ethics class in their education so far. AI, machine learning, big data, HCI, and cybersecurity courses were all mentioned as classes that asked students to think about ethics-adjacent topics such as the well-being of users or society. It was less common for classes in general software implementation or design, like software engineering or computer systems, to bring up ethics. Though "people" are inherently considered important in courses like cybersecurity and HCI, topics like AI, machine learning, and data science were also recognized as key areas for ethical consideration due to the widespread discussion about news coverage of the relevant ethical issues, as mentioned by P8-1:

> *"Since AI is trending now, there's a lot of work being done [...] Because it is having [a] larger impact. So I think only because of areas like NLP, deep learning, any of those areas – they are thinking of [the] negative impact. [...] But I think most of the courses don't [...] because in those areas, people are not thinking of any."*

However, for general software engineering, participants for the most part did not understand the relevance of ethics, and this affected how participants viewed the role of a software engineer. Many participants did not consider thinking about ethical impacts to be part of the process: when asked what their software engineering course taught, P2-2 stated:

> *"It concentrates on the process. [...] it actually trains us to be a software engineer, rather than concentrating on [the] impact [of] what we're going to build in this course."*

Others stated that ethical considerations were not a software engineer's job: *"the ethical implications, that's probably another department of the company, right?"* P10-1 mused. P7-2 stated that the people who ought to be responsible for ethical considerations were *"the designers, whoever they are, [they have] responsibility for* *the ethics stuff because they’re the ones who designed it to be good or bad. They’re the ones who can change it.”*

However, the overwhelming sentiment among participants was that in their educational experiences, ethics has not been covered enough or not well the majority of the time. For some project courses that did include ethical discussions, they would have at most a couple of lectures allocated to ethics, and several participants mentioned that the ambiguity around ethics discussions made it difficult to apply what they had learned. P11-2 reflected on their project course’s ethics lecture, stating:

> *“[The instructor] kind of made it [a] very subjective matter, so there is no right or wrong. [...] what we expect is like, tell us what is the right thing to do [...] I know it is subjective, but there are definitely some must do’s and must don’t’s. [...] Rather than just leaving it up in open space, like “you choose whatever you want to do.” That won’t teach us anything.”*

In the few courses that participants felt had effectively encouraged them to think about ethics, participants noted the frequency and applied nature of the discussions. P5-2 noted that in their computing group project course,

> *“[ethics] has been brought up in some of the lectures, like where they ask us to think of this topic where we have to think of what are we building, and how it’s affecting the user’s life. And there are some best practices also to ensure that our software is properly developed.”*

4.3 Participant Recommendations for Ethical Support

When asked what kind of support might be helpful if they were to include ethical considerations in their project work, participants overwhelmingly desired clarity and concrete information about ethics and how to create ethical software. As alluded to in the previous section, there was a desire to move away from open-ended and non-applied scenarios to more direct answers about good courses of action. Additionally, the desire for structure to explore ethics in computing subjects was often requested in the form of clear rules and guidelines to operate under. P4-1 stated,

> *“I think starting with a set of guidelines of how [...] not to negatively impact others. Having a structure of guidelines to prevent that would be useful while developing the project so that we know what we should not do.”*

Participants also wanted to learn more about current and past real world tech ethics issues, to better inform them of what to do if they encountered similar situations in the future. P6-2 suggested a repository of known issues: *“If we were taught some scenarios and case studies where we can learn ethics and then we can build that in our project, that would help a lot.”* P7-1 suggested that this repository could be compiled into *“a website reference with common pitfall solutions.”* Not only would this repository help with concrete, known issues students might face, but it would also spread awareness about the necessity of ethics knowledge; as stated by P4-2:

> *"Maybe sharing real world stories [...] like how ethics were violated previously in certain use cases or in certain user studies where people had to face lawsuits or something. [...] This happens in the world when you develop software. So maybe [students] don’t know how they want that, but the students must know that this is a thing. They can face such things in [the] real world.”*

Participants also mentioned wanting to learn specific computing techniques that could protect users. For example, P10-1 mentioned that there could be an upper division course that covered user safety: *“like a 4000 level [course titled] User Safety, that could all be about encryption [and] databases,”* while P4-2 suggested a course specific to software/product development: *“maybe we could get [a] specific course designed for these kind of things? Or ethical software development or ethical product development.”* The desire for clarity also manifested in requests for formal training; P9-1 suggested *“there should be some mandatory quiz or something to know what ethics is, so that people can think on their own, going forward.”*

Participants also expressed that feedback, especially from experts, would be valuable. The ideal experts would be either *“People who have gone through all [of the] developmental cycles of similar projects, [and] who know all the pitfalls that they went into”* as stated by P4-4, or *“someone who actually understands how modern technologies might impact, or how they affect or challenge certain ethics that we consider to be correct or true"* as suggested by P4-5. At the moment, a common and effective type of existing feedback in many participants’ classes came from professors and teaching assistants, often in the form of a project proposal review. P2-1 noted that ethics-related feedback could be part of that process:

> *“Like if we are proposing something, [...] if they say that there will be a lot of impact on people. Again, if you’re collecting the names and certificate number and stuff like that, we might not know how sensitive that information is, how it might affect the people. We’d be getting the feedback from the professor reviewing the project."*

In addition to formal feedback from course staff, participants frequently mentioned peer feedback via user testing to be valuable. P10-4 stated,

> *“I feel like during the actual coding process, just constant user testing would be nice just because then you get what the user feels and thinks. [...] So obviously an outside perspective would be nice because they’ll notice things that you personally didn’t.”*

Diverse perspectives were also mentioned as a key aspect of making user testing useful. P5-1 noted this:

> *“Once the software is ready, we can test it on a different group of people, to see how the users are behaving [...] using our software and what [...] they’re using [it] for. So like, using it on a different group of people might have [a] larger user set so that we can get to know the different user use cases of the software.”*

In terms of the software development lifecycle, participants suggested that the most pragmatic and effective times to introduce thinking about ethics were either very early in the design process (prior to starting the project at all, during the requirement gathering phase, or while designing), or soon before deploying to general users (during the testing phase). Thinking about ethics prior to starting the project was critical for P5-1, who noted that the fundamentals of a project cannot be changed very well once development has started: *"It is really a good thing to consider the impact of the finished project before. Later, the base of the project can't be changed, no matter what features we had."* Thinking about ethics while gathering requirements is important, as that is when developers can better understand what exactly the intended users are looking for. As noted by P1-1,

> *"Mainly the requirements will have a clear picture of [...] what the end goal is. And yeah, these questions come into [the] picture, how is it impacting a user who will be using it? In that case, we'll have a clear answer for all those things because we would've thought about it and then designed the project. Because the end goal is that we need to give [it] to a user, or when you're deploying it, that's what matters a lot."*

Participants also noted that thinking about ethics during the design phase would be easier than while in the middle of writing software for the project: P2-1 stated,

> *"It is important to actually think of how the finished project would be, how it might impact right in the development stages. [...] If we build it out and then see, [...] we might not have thought of something which might have actually caused impact, then again we have to like go back and do it [...] again if we keep thinking on what the end goal is and if we keep building on that, I think that would be more useful in building software."*

Specific helpful activities that participants suggested were (without using these terms) ethical speculation and adversarial thinking. P9-2 brainstormed the use of ethical speculation, stating *"So whatever idea that I think of, I also think of the alternate way, how it can be bad, like for each thing that I do, how can it fail?"* The cybersecurity concept of adversarial thinking was alluded to indirectly as writing user stories from the perspective of an "anti-user," described by P10-3 as:

> *"Maybe a use case test diagram to make it for the wrong purpose. [...] like you're trying to use the app as unethically as possible. Like people go in there and try to find the most unethical way to use your app. So you just know how to approach it when you're coding."*

Testing was viewed as a good time to think about ethics as well, including both user testing and technical testing suites. Testing was seen as a good option since it would catch last-minute bugs, and some participants seemed to consider ethical missteps as another type of bug. In the words of P8-1,

> *"In the end, [when] we do general testing in the software development [process], we also need to do testing for the negative impacts or ethics. [...] I think that should be something that is added to the lifecycle at the end for testing ethics."*

While it was not entirely clear what an ethical test suite would look like, participants drew parallels to other disciplines. P11-2 alluded to electrical engineering, where there are automated testing tools and external organizations to vet products:

> *"So what these people would do is, they kind of use a tool to build this design and check. Like are there any potential threats when it comes to cybersecurity? Not from [an] implementation perspective, but even before, like from a design perspective itself. [...] That tool would give them a list of threats and what these people would do is, they were [...] ranking them. [...] the approval this company makes, that would go to FTC for their approval on the equipment."*

In order for any of these methods to be useful, participants noted that there would need to be time baked into the class's project timeline to accommodate for thinking about ethics during design and testing. P6-2 noted that finishing computing projects was already tough: *"it gets difficult at one point. [...] you have just half a semester, or maybe just a couple of weeks to finish it up."* P3-2 noted that if some time could just be set aside and specifically allocated to concentrating on ethical considerations, thinking about ethics in computing group projects would be more feasible:

> *"I think without that, students would [...] probably just go straight into writing the code. So yeah, like having a set, like even an assignment where [...] the goal is just to design or to think about what methods [you are] going to use."*

Lack of time also plays a major role in participants' perceived lower priority of ethics in computing group projects. Some participants mentioned that in order to care, both in school and later in the workforce, they need a reason to prioritize it. As is common in an academic environment, they suggested this might come in the form of an impact on grades. In the words of P8-1, requirements for researching/stating ethical implications should be mandatory:

> *"I think [what] is needed in the assignments is something like what we had for our project, a rubric that has a specific point just for the impact. So I mean, at least having that people can think of it. [...] People know what can be the problems if they're given a chance, but if they are not asked to do it, people don't do it. So rather than support, I think it should be mandated. Seeing you will have to list down the impacts and how you are trying [...] to avoid those impacts."*

Other participants suggested a more lenient approach of making ethical deliberation extra credit, so that students would not be penalized, but professors could still indicate that ethical consideration is important. As stated by P6-1: *"it would be nice if [...] professors were like, okay, there would be bonus points if you [...] make [an] inclusive project or something."*

However, not all incentives have to be directly associated with grades, and this is an important distinction when contemplating solutions that can possibly scale to industry settings. One major incentive is redefining what it means to be a "good software engineer," as participants saw that as related to their job prospects. P10-5 noted that since considering ethical impacts is generally not seen as part of being a good computing professional at the moment, classes and therefore students do not prioritize it:

> *“I mean sure [computing courses] have ethics, the main thing is: to be a good programmer, do this. That’s why they wouldn’t probably touch on [...] any other problem that doesn’t really affect your look as a programmer I guess?”*

A specific way to redefine “good software engineers” such that the role includes ethics is to classify ethical concerns/features as functional requirements (features that must be implemented for the project to be considered “working”) rather than features that can be added after deploying initial versions of the software to users. As suggested by P8-2,

> *“I think ethics is mostly considered as a non-functional requirement. So we should actually consider it as a functional requirement. And you know, people would actually pay heed to it when it’s a functional requirement.”*

Several participants also noted that it is in the company/business’s best interest to make thinking about ethics part of being a good software engineer: P4-2 noted that pushing harmful or premature software *“can be detrimental to whatever company is pushing that tech or whatever community is pushing that tech. You lose the user’s trust and yeah, it’s very difficult to get that back.”*

5 DISCUSSION

Our goal in this study was to understand how ethics is currently a component in group project-based computing classes at our university, as well as how students would react to more direct engagement with ethics, and how such engagement might be supported. Our participants expressed that though some instructors are beginning to attempt incorporating ethics into group project courses, for the most part, the topic tends to fall on the backburner for both instruction and implementation. However, this is something that at least some students seem to actively want; one participant even thanked the researchers for raising these issues, which they had not been asked to think about previously.

One major contributing factor to participants not considering ethics on their own as part of their project work was that they do not perceive group projects as "real" – since e.g., there are not “real users” who could be harmed by the output of their project. Interestingly, prior work has shown that including ethics-related topics as part of technical course assignments can help students see the “real world” connections to what they are learning [24, 33]. It could very well be that both these things are true: students do not see the point of thinking about ethics because the project is not real, and also incorporating ethics could make a project seem more real. With this in mind, one potential solution could be for instructors to take steps to encourage students to tackle these projects as they might in an actual industry setting. It seems currently that students may not understand why it is important to approach class projects this way. We suggest that finding ways to emphasize that acting as if group projects are "real" will better prepare students for the workplace, encouraging students to take their projects more seriously and to consider the ramifications of their work.

Our findings also contribute to a growing body of research that points to ways that ethics in computing education in general could be improved. For our participants, the idea that software engineers are not responsible for ethical issues in technology and the concern that engineers are helpless in shaping product ethics in industry are still pervasive, but they also emphasized that they need to be better equipped with explicit resources when considering ethics for their work in school and on the job. For example, they frequently requested case studies and repositories of common tech ethics issues. These resources do exist; e.g., the ACM Code of Ethics [15], the AI Incident Database (incidentdatabase.ai), and other published sets of case studies [14, 37] provide examples and guidelines for talking about tech ethics controversies that instructors and students can potentially draw from.

Our findings also suggest that applied ethics in a specific computing context, as opposed to divorced from the technical content, can be very effective, which supports prior work suggesting the benefits of integrating ethics content into technical classes [7, 12, 20, 22]. However, a known challenge for instructors incorporating ethics is a lack of resources or knowledge [45]. Though resources do exist (including e.g. the Teaching Responsible Computing Playbook [36], publicly available syllabi [13], and assignments shared as part of published CS education research [7, 24, 26, 42]) and the availability of those resources can be very helpful to instructors interested in including ethics content in their courses [26, 45], the need for individualization to specific course content can be a significant challenge [6]. Brown et al. provide some suggestions for approaching creating ethics-related assignments for a specific class [6], and given that the structure of group project classes sometimes share similarities, instructors might consider collaborating with each other, a strategy that has been helpful for overcoming barriers to ethics integration [45].

Participants’ desire for explicit guidelines and rules also highlights another challenge in tech ethics education: striking a good balance between giving students specific strategies for tackling ethical issues and teaching them how to think through ambiguous ethical situations for themselves. At present, participants suggested that their own experiences with ethics education tended to lean toward the latter, being presented with ambiguous situations but rarely told what they would actually do about the issue in a real world scenario. On the other hand, in this context there may not always be a "right answer," particularly given the prevalence of ethical issues in emerging/novel technology, which means that teaching students *how* to think through problems may ultimately be more useful for them (even if that may not be obvious to the students). An analysis of syllabi for standalone tech ethics classes revealed that the most common learning objectives were around skills like critical thinking, issue spotting, and seeing multiple perspectives, rather than specific content [13]. We recommend that tech ethics pedagogy should be a mix of providing explicit guidance and teaching ways of thinking, since deliberating around ethical considerations is a skill, just like learning to code. Depending on the course content, instructors also might be able to draw from a growing body of research on concrete tools such as checklists and audits for considering ethics and fairness for topics like machine learning [16, 32, 39, 44], in addition to teaching strategies for tackling more ambiguous problems. It is important that instructors communicate to students *why* this is what they are learning, and how it will help them in the real world.

Additionally, given participants' interest in explicit assistance during project work, we see an opportunity for "ethics consultants." Participants frequently expressed the desire for feedback or help, often from someone more familiar with tech ethics than themselves, whether that be a professor or a peer. Having ethics consultants could fill that niche of ensuring feedback from an ethical lens. Participants elucidated some aspects of what such expert consultation would play: providing critiques on designs, user testing, and evaluating/constructing test cases that would catch “ethical bugs.” In order to do that, these consultants would need relevant technical background, awareness of social issues, and the ability to empathize with the many different kinds of users who utilize software during user testing. These experts might come in the form of teaching assistants, who can be an excellent resource especially if well-trained and supported [31], or even other students. Additionally, this type of assistance might mimic real world scenarios, since there are industry roles for people with applied tech ethics as a skillset [34], and industry professionals have expressed a desire for more collaboration with ethics domain experts [19]. Therefore, one skill that students might learn as well is how to know when to ask for help from someone with more expertise. Moreover, there is a role in computing education for training students for these kinds of specialist jobs, and so there might also be pedagogical benefit in having students help each other - for example, students taking standalone ethics courses serving as consultants for students in other courses.

Indeed, based on our findings we also think it is important to remember that improved ethics pedagogy in computing education cannot be a solution to problems of unethical practices and technology, without change in the tech industry as a whole. Participants shared the sentiment that school was just a stepping stone to their tech jobs, and cynicism over the state of the industry meant they were willing to forgo learning about ethics for skills they think will be more valued. This finding tracks to prior work that showed that students not only thought that ethics knowledge would not help them get a job, but might actively mean they would *not* get a job [40]. Additionally, results from the PEaRCE online learning system that simulated workplace scenarios for students also revealed that students were concerned about how bringing up ethical issues might impact their careers [9]. Moreover, research suggests that preemptive feelings of helplessness are not unfounded; even when software engineers are able to identify ethical issues, they may not have the power to resolve those concerns, due to e.g., workplace culture or organizational incentives [48]. Solutions to this overarching problem in many cases likely require significant structural changes.

Though it is also the case that industry has an opportunity to show that ethics is a priority, since computing coursework is often framed as preparation for the workforce – if the tech industry changes, education will likely follow suit. Specifically, changing what it means to be a “good software engineer” is highly dependent on industry: unless students and course staff are alerted that being a good software engineer requires thinking about ethics via job descriptions, interview questions, new ethical software engineering procedures, or some other means, ethics education may continue to be seen as a low priority for students. While the growing body of research on ethics education tends to frame the issue as creating change in universities to eventually instigate change in the broader tech industry via graduates, we note that the opposite approach of changing industry to affect computing education could have an impact as well.

Overall, our findings point to some potential best practices in designing a project-based computing course with ethics learning and practice in mind:

1. **Talk about ethics, justice, responsibility, and related topics early, often, and with context.** The computing courses our participants mentioned that were relatively more successful at incorporating ethics either brought up the topic near the beginning of the course (posing it as a fundamental part of the computing practice), and/or frequently talked about it when potential ethical issues arose around concepts directly related to the students' projects.
2. **Build ethical consideration into the timeline/structure of the group project.** When allocating timelines for computing group projects, leave some time to have students explicitly consider the impact and ethical implications of their work. That could look like asking students to think about the possible impacts of their work in their project proposal or initial designs (i.e. functional requirements around ethical impacts, adversarial thinking in user stories, etc), building in time for user testing after the initial project has been developed, and/or asking students to write test cases around possible negative impacts on users.
3. **Make ethics a priority for students.** Find a way to incentivize students to consider ethics while they code or design. One of the easiest ways to do this is to incorporate thinking about social impact into the rubric for the project assignment. The most ideal case would be to apply ethics into the very fabric of the technical workings, but even any analysis of potential ethical implications for a project can be helpful.
4. **Inform students why they should pretend this project is real and why ethics is an important part of that.** Students should understand that while the project isn't "real," the process is, and treating the project like it could have real world impact will help them be more prepared for the workforce. Make it clear that not only do they need to learn specific programming knowledge and project management skills, but they need to know what kinds of ethical issues they may face in industry, learn how to think about and mitigate those issues, and acknowledge the barriers that they may face in doing so.

It is challenging to make concrete recommendations since course designs and priorities are unique (which is indeed one of the challenges for creating ethics-integrated assignments [6]), though instructors could start by looking into existing resources for "plug and play" assignments and consider whether any of them might work for their needs. Some of these are designed for industry settings, which would be appropriate for group projects simulating that experience.

For example, there are a number of toolkits and exercises made specifically for helping technologists identify technological risks. The EthicalOS Toolkit includes foresight exercises and scenarios to kick off conversations around "visualizing and anticipating future risk of technology products, acknowledging that once technology is released and reaches scale it may be used for purposes beyond the original intention" [17]. The "Black Mirror Writers' Room" is a creative activity to help students speculate about future harms and consequences of current technologies [26], with the idea being that if students are engaged by imagining science fiction scenarios, the activity could scaffold them into ethical speculation for current projects they are working on. The Building Utopia Toolkit employs an Afrofuturist lens for speculative design processes in ways that can prompt conversation about race in design and the immediacy of social issues [5]. These types of exercises could be used or adapted to help scaffold students in explicitly thinking through possible unintended consequences of their projects.

There are other specific exercises and tools that might work well for different types of courses in ways that might, as requested by one of our participants, help turn ethics into functional requirements. For example, machine learning, data science, or AI courses might require students to use tools such as the RealML worksheet [44], datasheets for datasets [16], or dataset nutrition labels [21] as part of their documentation. UX and HCI courses that already require students to engage in the creation of user personas could require specific types of adversarial personas or personas for non-dominant or vulnerable users.

Instructors might also simply create a series of questions (tailored to make sense for their course) for students to reflect on. "What are the ethical implications of this project?" is likely far too broad for students to engage with, but more specific questions might include:

- Who is likely to have access to this technology, and what are the implications of lack of access?
- Are there ways in which this technology could result in someone's privacy being violated?
- Are there ways that potential harms, no matter how small, would be felt disproportionately by different groups of people?
- What impact would this technology have on the environment?
- How might a bad actor make use of this technology?
- What is the worst and best news headline that could be written about this technology?
- What are the ideal benefits of this technology and how can we ensure that those benefits are felt equally?

Finally, we want to emphasize that a component of making ethics more "real" in terms of students' future practice is an acknowledgment of the barriers that they may face. In articulating the important role that *power* plays in limiting tech professionals' ability to resolve ethical concerns, Widder et al. point out that educational interventions that stop at issue spotting will leave students unprepared [48]. We suggest that group projects, but especially capstone classes that are explicitly designed to mimic real world product design, are the ideal context to teach relevant skills. This might include discussion of, as Widder et al. identify, negotiating for ethics in the context of organizational incentives, calling attention to strategies for building collective power, and the role of whistleblowing [48]. In terms of concrete exercises that can be built into projects, we would encourage instructors to require students to not only identify potential ethical issues, but to make persuasive arguments for them to be resolved. Once you answer the question "What could go wrong?" how would you bring that issue to someone in the position to fix it? Computer science students often do not receive adequate training in communication skills [11], and practice in such persuasive arguments could be an incredibly important part of simulating real world practice, alongside technical skills. Moreover, students should be taught about and encouraged to reflect on what they can do once they are in the workforce to help create workplace cultures conducive to responsible tech.

6 CONCLUSION AND FUTURE WORK

This work takes an initial look at the role of ethics education in group projects, and points to ways in which ethics might be more effectively integrated into coursework and the software engineering process. Yet, it only scratches the surface of what can be promising pedagogical strategies. As mentioned, one limitation of this study is that the focus groups were conducted on a narrow sample. A good direction for future work would be to replicate this study and compare insights on a larger scale and in different educational settings. Mapping insights from students at different universities to their curricula would help characterize effective features of ethics education, in addition to increasing the brainstorming potential for ideas and strategies for improving ethics education. We also hope that instructors will consider trying some of the recommendations based on this research as well as thinking through other ideas more specific to the context of their classes, and potentially to evaluate interventions as well. Finally, we emphasize that part of what can make "real world" experiences simulated in the classroom better at preparing students for their careers is to acknowledge the barriers they may face, and to discuss ways to work towards improving the tech industry as a whole. Group projects have the potential to become more than just an effective way to learn about code or project management – they can be a way for students to experience the responsibility and impact technologists have on society as well.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation, IIS Award #2046245.

REFERENCES

[1] Genevera I. Allen. 2021. Experiential Learning in Data Science: Developing an Interdisciplinary, Client-Sponsored Capstone Program. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 516–522.

[2] Priyatham Anisetty and Paul Young. 2011. Collaboration Problems in Conducting a Group Project in a Software Engineering Course. *Journal of Computing Sciences in Colleges* 26, 5 (2011), 45–52.

[3] Angela Bielefeldt, Madeline Polmear, Daniel Knight, Nathan Canney, and Christopher Swan. 2017. Incorporation of Ethics and Societal Impact Issues into Senior Capstone Design Courses: Results of a National Survey. *ASEE Annual Conference & Exposition* (2017).

[4] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[5] Kirsten E Bray, Christina Harrington, Andrea G Parker, N'Deye Diakhate, and Jennifer Roberts. 2022. Radical futures: Supporting community-led design engagements through an afrofuturist speculative design toolkit. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*. 1–13.

[6] Noelle Brown, Korrian South, Suresh Venkatasubramanian, and Eliane S. Wiese. 2023. Designing Ethically-Integrated Assignments: It's Harder Than it Looks. *Proceedings of the ACM ICER International Computing Education Research Conference* (2023).

[7] Noelle Brown, Koriann South, and Eliane S Wiese. 2022. The Shortest Path to Ethics in AI: An Integrated Assignment Where Human Concerns Guide Technical Decisions. In *Proceedings of the ACM ICER Conference on International Computing Education Research*. 344–355.

[8] Noelle Brown, Benjamin Xie, Ella Sarder, Casey Fiesler, and Eliane S Wiese. 2024. Teaching Ethics in Computing: A Systematic Literature Review of ACM Computer Science Education Publications. *ACM Transactions on Computing Education* 24, 1 (2024), 1–36.

[9] Francisco Castro, Sahitya Raipura, Heather Conboy, Peter Haas, Leon Osterweil, and Ivon Arroyo. 2023. Piloting an Interactive Ethics and Responsible Computing Learning Environment in Undergraduate CS Courses. *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education* (2023), 659–665.

[10] Lena Cohen, Heila Precel, Harold Triedman, and Kathi Fisler. 2021. A new model for weaving responsible computing into courses across the CS curriculum. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 858–864.

[11] Katrina Falkner and Nickolas JG Falkner. 2012. Integrating communication skills into the computer science curriculum. In *Proceedings of the ACM SIGCSE Technical symposium on Computer Science Education*. 379–384.

[12] Casey Fiesler, Mikhaila Friske, Natalie Garrett, Felix Muzny, Jessie J Smith, and Jason Zietz. 2021. Integrating ethics into introductory programming classes. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 1027–1033.

[13] Casey Fiesler, Natalie Garrett, and Nathan Beard. 2020. What do we teach when we teach tech ethics? A syllabi analysis. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 289–295.

[14] Markkula Center for Applied Ethics. [n. d.]. Ethics Cases. https://www.scu.edu/ethics/ethics-resources/ethics-cases/

[15] Association for Computing Machinery. [n. d.]. ACM Code of Ethics - Case Studies. https://www.acm.org/code-of-ethics/case-studies

[16] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM* 64, 12 (2021), 86–92.

[17] Paula Goldman and Raina Kumra. 2018. Introducing the (World's First) Ethical Operating System. *Omidyar Network* (2018). https://medium.com/omidyar-network/introducing-the-worlds-first-ethical-operating-system-7acc4abc2bfa

[18] Holly Golecki and Joe Bradley. 2023. Experiential Learning: Exploring Nuances When Making Ethical Decisions in a Capstone Design Course. *Biomedical Engineering Education* (2023), 1–8.

[19] Tricia A Griffin, Brian P Green, and Jos VM Welie. 2024. The ethical wisdom of AI developers. *AI and Ethics* (2024), 1–11.

[20] Barbara J Grosz, David Gray Grant, Kate Vredenburgh, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. 2019. Embedded EthiCS: integrating ethics across CS education. *Commun. ACM* 62, 8 (2019), 54–61.

[21] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2020. The Dataset Nutrition Label. *Data Protection and Privacy* 12, 12 (2020).

[22] Diane Horton, David Liu, Sheila A. McIlraith, Steven Coyne, and Nina Wang. 2024. Do Embedded Ethics Modules Have Impact Beyond the Classroom?. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education V. 1*. 533–539.

[23] Diane Horton, Sheila A. McIlraith, Nina Wang, Maryam Majedi, Emma McClure, and Benjamin Wald. 2022. Embedding Ethics in Computer Science Courses: Does It Work?. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education - Volume 1*. 481–487.

[24] Julie Jarzemsky, Joshua Paup, and Casey Fiesler. 2023. " This Applies to the Real World": Student Perspectives on Integrating Ethics into a Computer Science Assignment. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 374–380.

[25] Mara Kirdani-Ryan and Amy J Ko. 2022. The house of computing: Integrating counternarratives into computer systems education. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 279–285.

[26] Shamika Klassen and Casey Fiesler. 2022. " Run Wild a Little With Your Imagination" Ethical Speculation in Computing Education with Black Mirror. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education-Volume 1*. 836–842.

[27] Amy J Ko, Alannah Oleson, Neil Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It is time for more critical CS education. *Commun. ACM* 63, 11 (2020), 31–33.

[28] Matthew Le Bui and Safiya Umoja Noble. 2020. We're missing a moral framework of justice in artificial intelligence. *The Oxford handbook of ethics of AI* (2020), 163–179.

[29] Yue Li, Soon Phei Tin, and Charlie Reis. 2022. Student Engagement in Software Engineering Group Projects: An Action Research Study. In *Proceedings of the 2022 5th International Conference on Education Technology Management*. 367–374.

[30] Kevin Lin. 2022. CS education for the socially-just worlds we need: The case for justice-centered approaches to CS in higher education. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 265–271.

[31] Robert MacDonald, Cass Zegura, Ben Rydal Shapiro, Jason Borenstein, and Ellen Zegura. 2023. Developing Community Support for Computing Ethics Teaching Assistants. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 360–366.

[32] Michael A. Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. 2020. Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*. 1–14.

[33] Nora McDonald, Adegboyega Akinsiku, Jonathan Hunter-Cevera, Maria Sanchez, Kerrie Kephart, Mark Berczynski, and Helena M Mentis. 2022. Responsible Computing: A Longitudinal Study of a Peer-led Ethics Learning Framework. *ACM Transactions on Computing Education (TOCE)* 22, 4 (2022), 1–21.

[34] Jacob Metcalf, Emanuel Moss, et al. 2019. Owning ethics: Corporate logics, silicon valley, and the institutionalization of ethics. *Social Research: An International Quarterly* 86, 2 (2019), 449–476.

[35] Sriram Mohan, Stephen Chenoweth, and Shawn Bohner. 2012. Towards a Better Capstone Experience. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 111–116.

[36] Mozilla. [n. d.]. https://foundation.mozilla.org/en/what-we-fund/awards/teaching-responsible-computing-playbook/. https://foundation.mozilla.org/en/what-we-fund/awards/teaching-responsible-computing-playbook/

[37] MIT Schwarzman College of Computing. [n. d.]. Case Studies in Social and Ethical Responsibilities of Computing. https://www.scu.edu/ethics/ethics-resources/ethics-cases/

[38] Aadarsh Padiyath. 2024. A Realist Review of Undergraduate Student Attitudes towards Ethical Interventions in Technical Computing Courses. *ACM Transactions on Computing Education* (2024).

[39] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the ACM FAccT Conference on Fairness, Accountability, and Transparency*. 33–44.

[40] Ella Sarder and Casey Fiesler. 2022. Entering the Techlash: Student Perspectives on Ethics in Tech Job Searches. In *Companion Publication of the ACM CSCW Conference on Computer Supported Cooperative Work and Social Computing*. 85–88.

[41] Irving Seidman. 2006. *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. Teachers college press.

[42] Ben Rydal Shapiro, Emma Lovegall, Amanda Meng, Jason Borenstein, and Ellen Zegura. 2021. Using role-play to scale the integration of ethics across the Computer Science curriculum. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 1034–1040.

[43] Michael Skirpan, Nathan Beard, Srinjita Bhaduri, Casey Fiesler, and Tom Yeh. 2018. Ethics education in context: A case study of novel ethics activities for the CS classroom. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 940–945.

[44] Jessie J Smith, Saleema Amershi, Solon Barocas, Hanna Wallach, and Jennifer Wortman Vaughan. 2022. Real ml: Recognizing, exploring, and articulating limitations of machine learning research. In *Proceedings of the ACM FAccT Conference on Fairness, Accountability, and Transparency*. 587–597.

[45] Jessie J Smith, Blakeley H Payne, Shamika Klassen, Dylan Thomas Doyle, and Casey Fiesler. 2023. Incorporating Ethics in Computing Courses: Barriers, Support, and Perspectives from Educators. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 367–373.

[46] Norman Makoto Su, Amanda Lazar, and Lilly Irani. 2021. Critical affects: Tech work emotions amidst the techlash. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–27.

[47] Sander Valstar, Caroline Sih, Sophia Krause-Levy, Leo Porter, and William G Griswold. 2020. A quantitative study of faculty views on the goals of an undergraduate CS program and preparing students for industry. In *Proceedings of the ACM ICER Conference on International Computing Education Research*. 113–123.

[48] David Gray Widder, Derrick Zhen, Laura Dabbish, and James Herbsleb. 2023. It's about power: What ethical concerns do software engineers have, and what do they (feel they can) do about them? *Proceedings of the ACM FAccT Conference on Fairness, Accountability, and Transparency* (2023), 467–479.

Received 29 March 2024